

多處理機平行程式設計作業 2

F74096116 曾瑜華

Problem1:

1. What have you done?

第一題基本上是照著作業的實作參考方法去做，先將影像以不同 row 的方式分割給不同的 process，接著將各自的影像矩陣以 Scatterv 傳送給其他 process。然後以每個分割空間的最上層 row 和最下層 row 當作暫存空間去拿到模糊計算所需的九宮格。處理完資料就進行平滑運算，結束後再 Gatherv 到 process 0，得到結果。

2. 測試結果

1.3 Process 數量	1.2 執行時間(s)
1	113.128
8	51.448
16	40.94
32	46.452
40	48.188

可以看到當 process 數目為 1，即為未平行化時的時間最久，而隨著 process 數目的上升，執行時間也隨之下降，待到 process 來到 32 時，overhead 造成的影響已經讓時間不減反增。

接著固定 process 為 16，測試不同的平滑處理數目。

i2 ₃ 平滑次數	1.2 執行時間(s)
10	0.52
100	3.656
1000	36.34
10000	438.252

可以發現隨著平滑次數的上升，執行時間也不斷的上升，甚至是接近線性成長的結果。

3. Any difficulties?

這次用到的 MPI 指令比起上次增加了許多，因此在應用上面需要不斷的去尋找 documents，甚至就連寫完之後，都還不確定自己到底理解了沒。還有和上次的題目相比，題目的應用性更高，不再只是單純讓我們練習 MPI 而已。

Problem2:

1. What have you done?

第二題一樣參照作業上提供的寫法，而這題我是以 C++ 來完成的，首先在 process 0 輸入陣列的大小，接著 broadcast 給其他 process，再來利用 rand 函式給他們產生亂數，然後讓每個 process 去平均分配陣列，而沒有分配完的部分再分散到 rank 較小的 process 內。然後使用 C++ 內建的 sort 函式將每個 process 內的陣

列先排序好，再去應用上 odd-even sort，最後再將結果 Gatherv 到 process 0。

```
Please enter the number of size to sort: 10
Process 0local array
40109922 116310668 207806424 497991097 846085262 950627833 1485461953 1672193589 1822110912 1905861564
Total execution time is: 3.6302e-05
The sorted array: 40109922 116310668 207806424 497991097 846085262 950627833 1485461953 1672193589 1822110912 1905861564
```

可以看到執行結果的 output 都符合題目要求，後面由於數字太大，故先將 local array 以及 answer array 的 print 都先 command 掉。

2. 執行結果

陣列大小	Process數量	1.2 執行時間
1000	1	0.000953114
1000	8	0.444633
1000	16	0.808152
1000	32	1.19599
10000	1	0.0471843
10000	8	0.364
10000	16	0.764025
10000	32	1.28403
100000	1	0.0164949
100000	8	0.275978
100000	16	0.757893
100000	32	1.23276
1000000	1	0.438322
1000000	8	1.46734
1000000	16	2.6434
1000000	32	3.56003

可以看到，即便 process 數量有上升，但沒有一個的 performance 比未平行化的好。猜測是因為在 sort 的部分，由於是未平行化，所以基本上就等於在使用 C++ 內建的 sort，時間複雜度為 $O(n\log n)$ ，而 sort 的計算量

也沒有到很大，因此溝通之間的 overhead 所造成的影響遠遠超過平行化帶來的效益。

3. Any difficulties?

和第一題一樣，要先去了解 Scatterv 和 Gatherv 的參數已經使用方法，還有 odd-even sort 的運作方法，做完這些難題基本就能做完這題。

Feedback:

上次的作業還勉強能應付，這次的作業就對我來說非常的難了，不知道自己是不是在這方面沒有天賦，但可以看出題目助教還是出的非常用心，而且也不斷的在 DC 回答同學的問題，辛苦助教了！